SISTEMAS OPERATIVOS

MANUAL DE USUARIO

SIMULACIÓN GRÁFICA DE UN COMPUTADOR FICTICIO DE FUNCIONAMIENTO BÁSICO “CHMÁQUINA”



CONTENIDO

OBJETIVO

Esta aplicación tiene como objetivo principal permitir la realización de la simulación de un computador ficticio de funcionamiento básico al cual se denominará **CHMÁQUINA.** Ello se realiza a partir de la carga de algunos programas que posteriormente se compilan y se ejecutan permitiendo que puedan conocerse los respectivos resultados; es posible visualizar cada línea de instrucción que está siendo ejecutada, el valor que adquiere el acumulador y el número del programa que está en ejecución.

Se presenta por parte del **CHMÁQUINA** un ambiente de simulación gráfica de un computador ficticio, que comprende un espacio determinado de memoria, el núcleo del sistema operativo, procesador, componentes como monitor e impresora que permiten visualizar los resultados que se obtienen y otros elementos que también apoyan el proceso de visualización de la ejecución de un programa.

Para facilitar el uso de esta aplicación el código del lenguaje empleado implica poseer conocimientos básicos de programación, así como de la arquitectura del computador, con el fin de comprender como funciona cada proceso. El **CHMÁQUINA** permite conocer si existen errores en caso de haberlos con el fin de depurarlos, y así proceder a realizarse la carga definitiva del programa en memoria para su ejecución.

Las siguientes son instrucciones que corresponden al código de operación que pueden ser ejecutadas en el programa:

- **CARGUE:** Cárguese/copie en el acumulador el valor almacenado en la variable indicada por el operando.

- **ALMACENE:** Guarde/copie el valor que hay en el acumulador a la variable indicada por el operando.

- **VAYA*:** Salte a la instrucción que corresponde a la etiqueta indicada por el operando y siga la ejecución a partir de allí.

- **VAYASI*:** Vaya a la instrucción que corresponde a la etiqueta indicada por el primer operando si el valor del acumulador es mayor de cero, a la instrucción que corresponde a la etiqueta indicada por el segundo operando si el valor del acumulador

es menor a cero o a la siguiente instrucción adyacente a la instrucción **vayasi** si el acumulador es cero y siga la ejecución a partir de allí.

- **NUEVA:** Crea una nueva variable cuyo nombre es el especificado en el primer operando, en el segundo operando definirá el tipo de variable (**C** Cadena/alfanumérico, **I** Entero, **R** Real/decimal), un tercer operando establecerá un valor de inicialización; a cada variable se le asignará automáticamente una posición en la memoria. Las variables deberán estar definidas antes de ser utilizadas. Las variables no inicializadas tendrán por defecto el valor cero para reales y enteros y espacio para cadenas. El separador de decimales es el punto.

- **ETIQUETA:** Crea una nueva etiqueta cuyo nombre es el especificado en el primer operando y a la cual le asignará automáticamente la posición indicada en el segundo operando (este será la posición relativa a la primera instrucción del programa). Las instrucciones que definen etiquetas podrán definirse en cualquier posición del programa, pero en todo caso antes de que la instrucción retorne.

- **LEA:** Lee por teclado el valor a ser asignado a la variable indicando por el operando.

- **SUME:** Incremente el valor del acumulador en el valor indicado por la variable señalada por el operando.

- **RESTE:** Decrementa el acumulador en el valor indicado por la variable que señala el operando.

- **MULTIPLIQUE:** Multiplique el valor del acumulador indicado por la variable señalada por el operando.

- **DIVIDA:** Divida el valor del acumulador por el valor indicado por la variable señalada por el operando. El divisor deberá ser una cantidad diferente de cero.

- **POTENCIA:** Eleve el acumulador a la potencia señalada por el operando (los exponentes deben ser valores enteros, positivos o negativos).

- **MODULO:** Obtenga el modulo al dividir el valor del acumulador por el valor indicado por la variable señalada por el operando.

- **CONCATENE:** Genere una cadena que una la cadena dada por el operando a la cadena que hay en el acumulador.

- **ELIMINE:** Genere una subcadena que elimine cualquier aparición de los caracteres dados por el operando de la cadena que se encuentra en el acumulador.

- **EXTRAIGA:** Genere una subcadena que extraiga los primeros caracteres (dados por el operando) de la cadena que se encuentra en el acumulador.

- **MUESTRA:** Presente por pantalla el valor que hay en la variable por el operando, si el operando es acumulador muestre el valor del acumulador.

- **IMPRIMA:** Lo mismo que el anterior pero presentándolo en la impresora.

- **RETORNE:** El programa termina; debe ser la última instrucción del programa y tiene como operando opcional un valor numérico de 0 a 255.

PARA RECORDAR: El código puede tener comentarios por líneas, estos van denotados por dos backslash (**//**), los cuales deben ir al inicio de la línea. Es importante tener en cuenta que todo lo que se encuentre después de estos símbolos hasta el fin de línea, no se toma en cuenta a la hora de la compilación y ejecución del programa, sin embargo ello no quiere decir que no ocupen un espacio en la memoria.

El programa no permite la sobrecarga del acumulador (Overflow) por lo cual es posible que el funcionamiento de este no sea el esperado. Todas las instrucciones se escriben en letra minúscula y las líneas de los programas empiezan en cero.

PANTALLA PRINCIPAL DEL CHMÁQUINA

Desde el momento en el que se accede a la aplicación es posible visualizar gracias a la pantalla principal las diferentes opciones con las que cuenta el CHCOMPUTADOR ficticio para su respectiva configuración.

A partir de esta se permite la carga de los programas con extensión .ch, su posterior compilación, depuración de errores y modificación de estos (a partir de la opción salvar).

Se permite además que la ejecución de los programas pueda darse de manera continua o en ejecuciones paso a paso.

Cuando se realizan modificaciones como pueden ser la carga de un nuevo programa en la memoria, o el nuevo valor de una variable o el acumulador, la memoria se actualiza.

Las siguientes son las diferentes opciones con las que cuenta la pantalla inicial de la aplicación y su descripción:

1. **ARCHIVO:** Cuenta con opciones para abrir y salvar los programas .ch.

2. **COMPILAR:** Se compila y ejecuta en modo continuo el programa.

3. **VISUALIZAR:** Permite mostrar la memoria.

4. **EJECUCIÓN:** Se compila y ejecuta el programa paso a paso, puede ejecutarse, pararse su ejecución y/o reanudarse.

5. En este espacio se muestra el número del programa que está en ejecución, la instrucción (PC) y el valor que adquiere el acumulador.

6. Es un espacio asignado que hace la función de una impresora, allí se da la instrucción “imprima”.

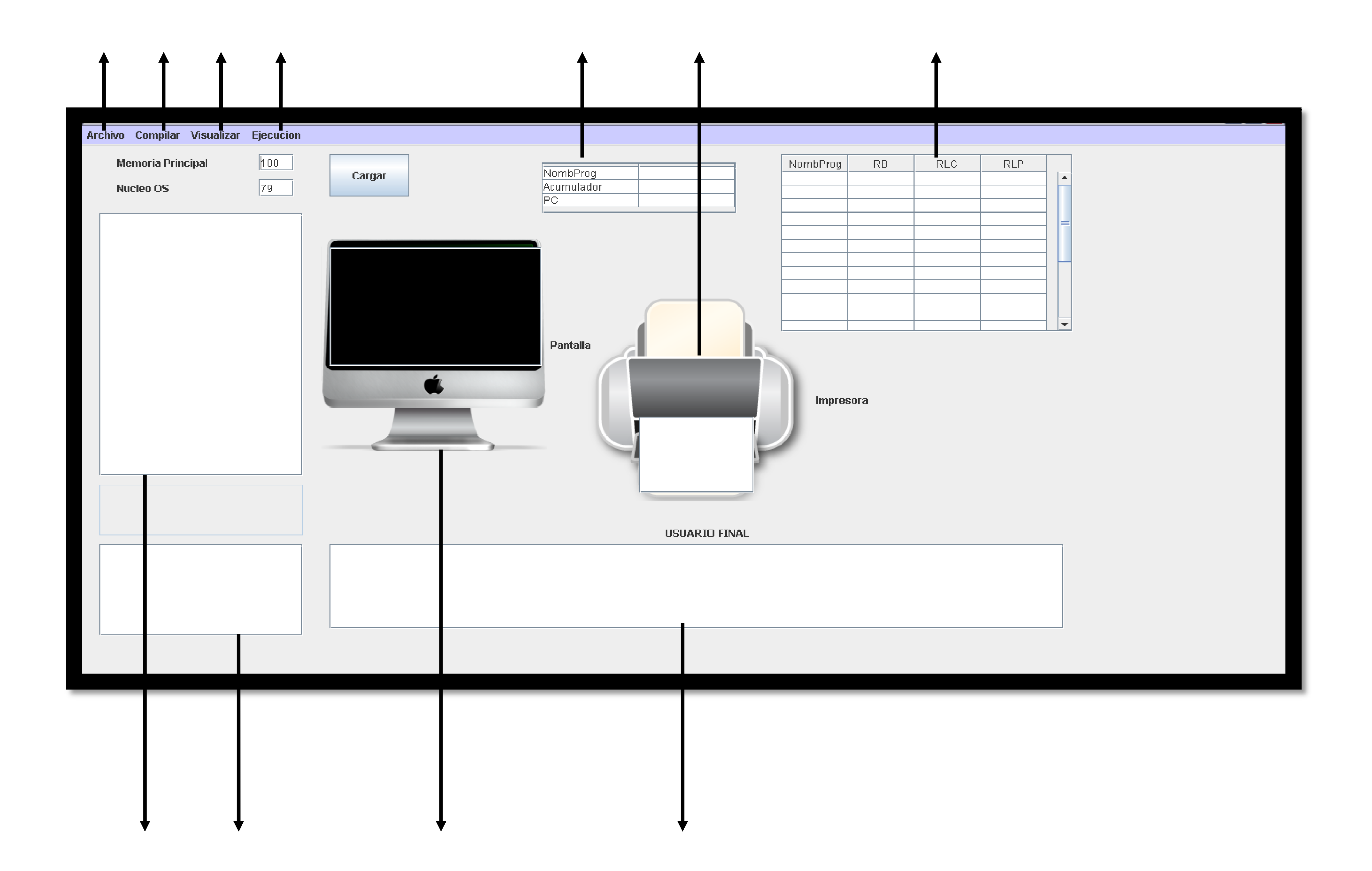
7. En este espacio se registra la información de los programas que son cargados en memoria, la información que se maneja es: Número de programa, Registro base (RB que indica en donde empieza el programa en memoria), Registro límite de código (RLC que indica hasta donde llega el programa) y el Registro límite de programa (RLP que presenta la última instrucción del programa).

8. **Áreas de programa o trabajo**

9. **Áreas de errores**

10. Es un espacio asignado que hace la función de un monitor, allí se da la instrucción “muestre”.

11. Este espacio permite mostrar la memoria del computador.

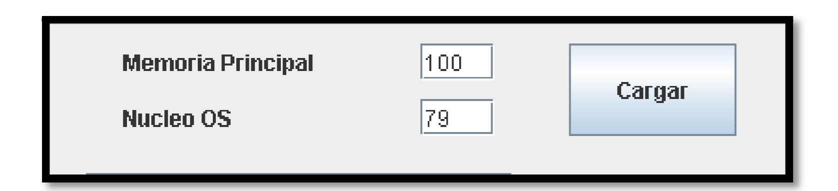


Archivo
Compilar
Visualizar
Ejecucion
Memoria Principal
100
Nucleo OS
79
Cargar
NombProg
Acumulador
PC
NombProg
RB
RLC
RLP
Pantalla
Impresora
USUARIO FINAL

INSTRUCCIONES DE MANEJO DEL “CHMÁQUINA”

- **SELECCIONAR VALORES PARA LA MEMORIA Y EL SISTEMA OPERATIVO**

En el momento de acceder a la aplicación es posible reconocer unos campos que están asignados para la memoria principal a través de un vector de hasta 99999 posiciones las cuales pueden ser variadas al momento de iniciar el programa; inicialmente el chcomputador cuenta con 100 posiciones para la memoria, las primeras posiciones de la memoria estarán reservadas para el núcleo del sistema operativo, e inicialmente tiene un valor por defecto de 10*z + 9 posiciones (donde z es una condición del último digito de cédula del estudiante encargado del proyecto). En caso de escoger estos valores o de seleccionar otros es necesario utilizar la opción “**Cargar**”



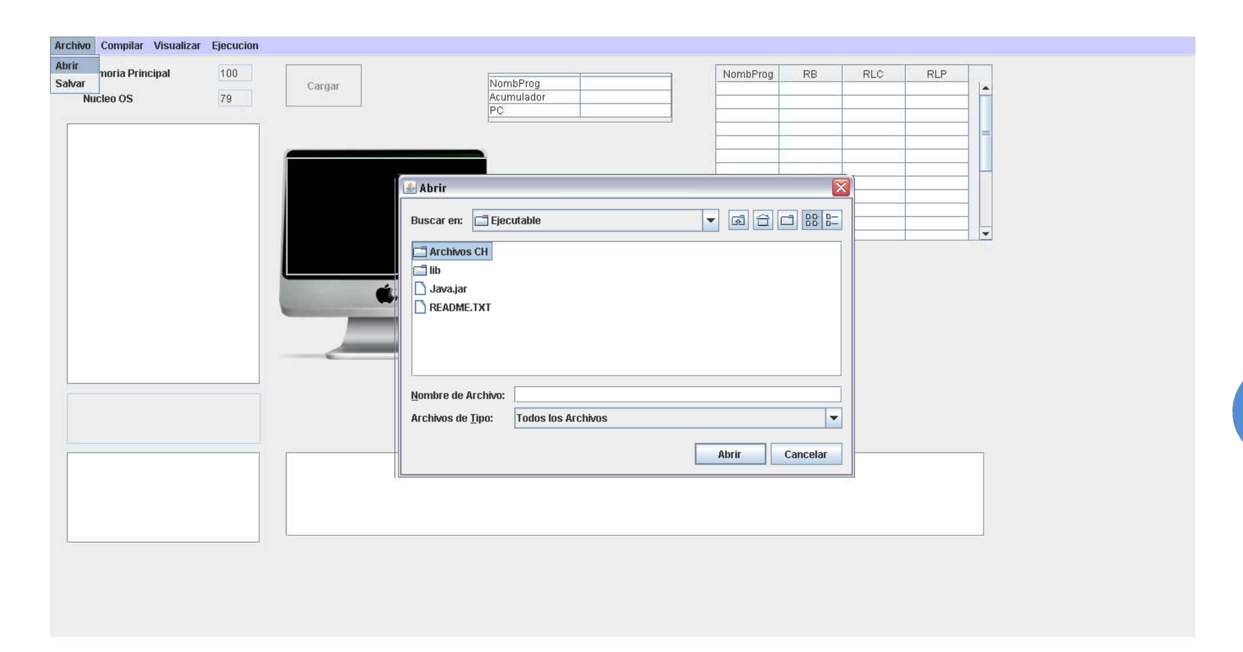
NOTA: Es importante tener en cuenta que el tamaño del núcleo del sistema operativo no sea superior al de la memoria.

- **SELECCIONAR EL ARCHIVO**

Al momento de seleccionar el menú “**Archivo**” es posible reconocer dos opciones:

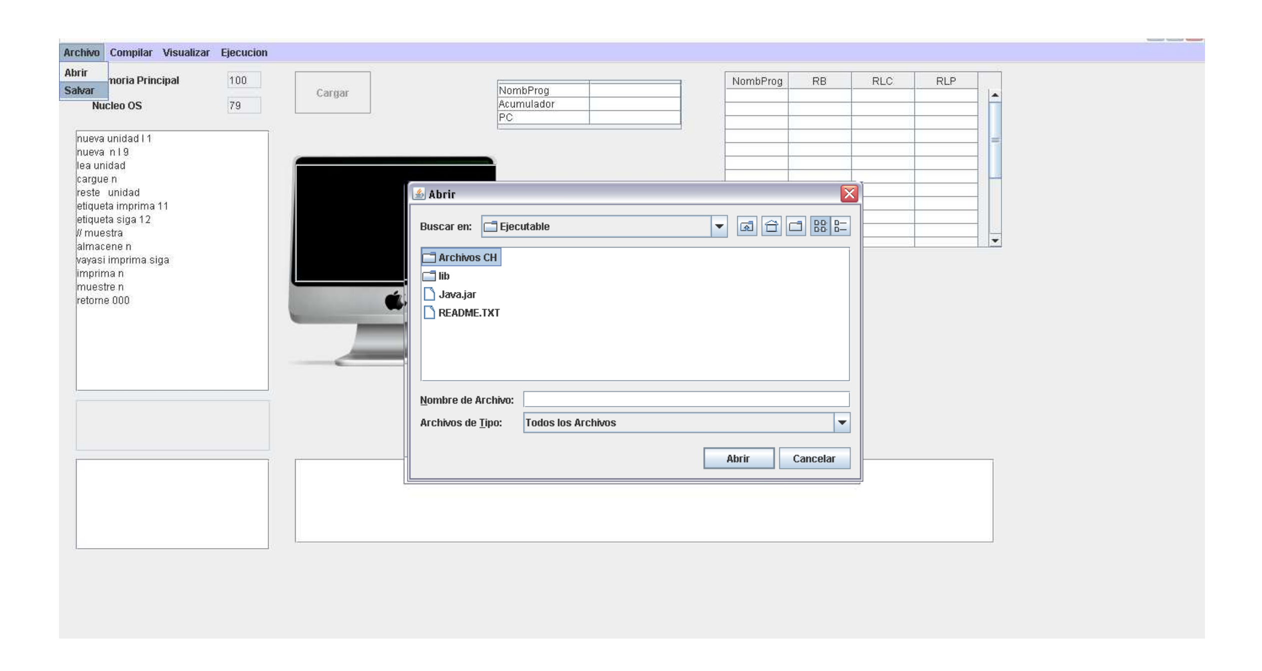
a) El archivo puede subirse desde cualquier carpeta que contenga chprogramas, los cuales deberán haber sido almacenados con la extensión .ch.

>> Archivo >> Abrir



b) El archivo podrá ser creado en el área reservada para trabajo, este será un chprograma el cual podrá ser almacenado en cualquier carpeta del computador.

>> Archivo >> Salvar

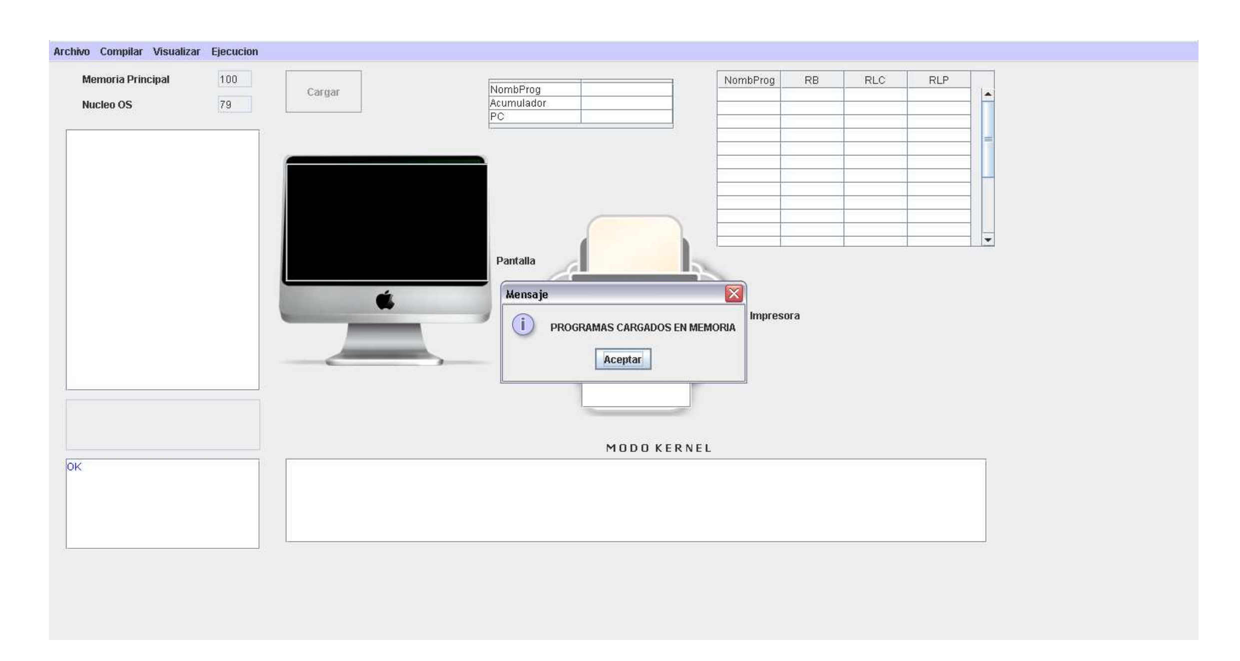


- **COMPILACIÓN DEL ARCHIVO**

Cuando el archivo se encuentre en el área de texto, es necesario compilarlo para realizar una verificación de la sintaxis y producir así una lista con los errores que estén presentes en el código.

En el área designada para los errores, estos son presentados en forma descendente al programa, se muestra cuál ha sido el tipo de error y se destaca la línea en la cual ha sido encontrado.

NOTA: La aplicación no toma en cuenta los espacios en blanco y las líneas del código son contadas a partir de cero.



Cuando los programas seleccionados estén libres de errores, se realiza su carga en la memoria para posteriormente realizar su ejecución.

NOTA: A los programas que se cargan y a sus variables se les asigna un espacio.

En el momento en que se carguen varios programas en el área de trabajo y se desee compilarlos, se comprueba que todos ellos estén libres de errores para proceder a cargarlos en la memoria. Finalmente cuando un programa esté libre de errores se borra de esta área y se carga en la memoria, dejando que los demás programas continúen con el proceso de compilación.

NOTA: La carga de los programas en la memoria se realiza de manera separada para cada uno.

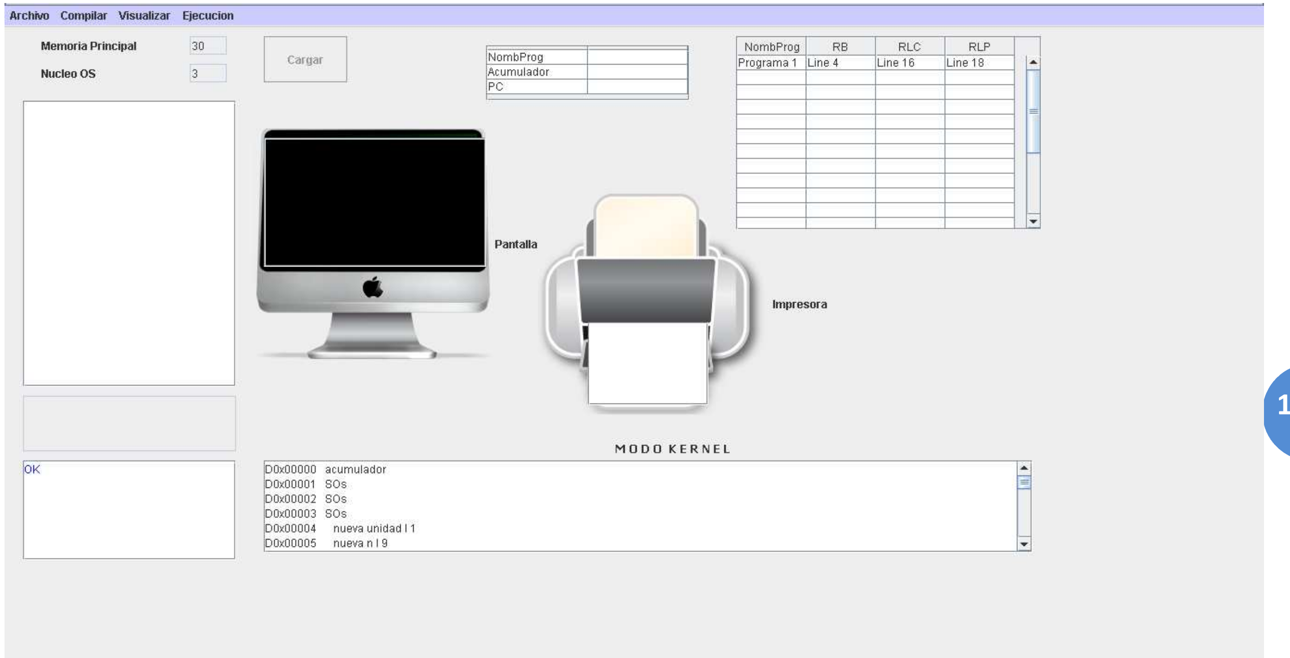
TIPOS DE ERRORES QUE SE PUEDEN PRESENTAR

- ✓ **Línea fuera de los límites del programa:** Este error surge principalmente cuando una de las líneas de una etiqueta supera en tamaño las de un programa.
- ✓ **Incompatibilidad de tipos:** Es un error que puede presentarse fácilmente debido a la asignación de valores que no corresponden al tipo de una variable.
- ✓ **Operandos incompletos y operandos "extra :** Se presentan cuando se realiza una errónea sintaxis de las operaciones manejadas en el lenguaje ch, y los operandos "extra" se dan cuando una instrucción maneja más operandos de los que están permitidos.
- ✓ **Último programa sin la sentencia "retorne":** Hace que sea imposible reconocer en donde finaliza el programa.
- ✓ **Variable no encontrada:** Es necesario declarar una variable para poderla utilizar en cualquier operación.
- ✓ **Nombres de variables no autorizados:** Es necesario que las variables manejen nombres diferentes a los que son reservados para las diferentes operaciones del programa (sume, reste, multiplique, divida, imprima, retorne….).

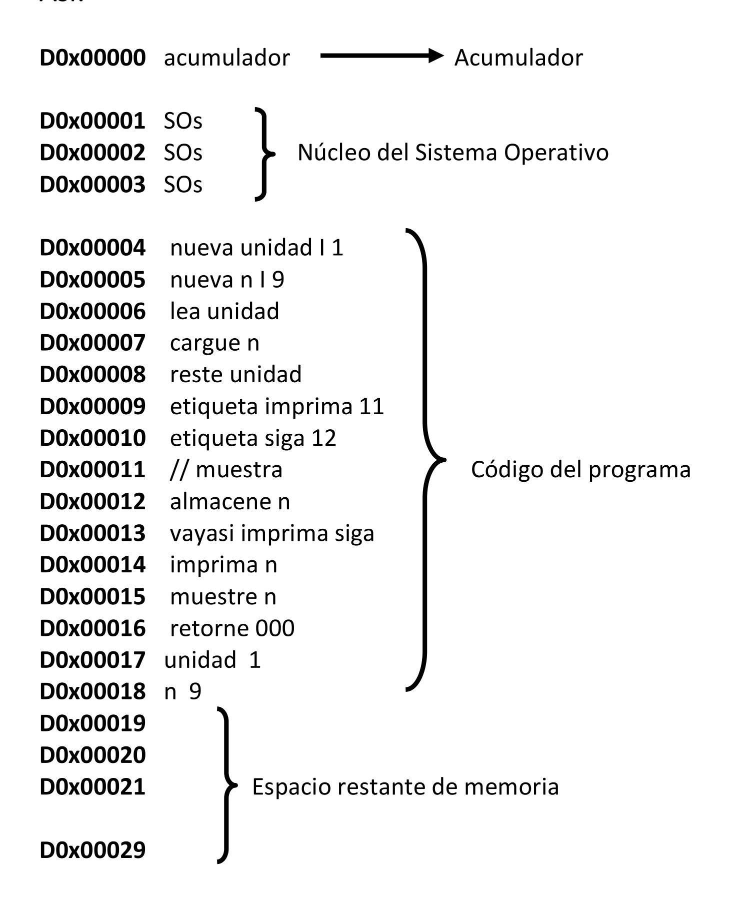
- **VISUALIZACIÓN DE LA MEMORIA**

Si se desea reconocer como queda la memoria después de la carga de los programas es posible elegir la opción del menú denominada "**Visualizar**".

En otro caso sólo será posible cuando comience a ejecutarse el programa.



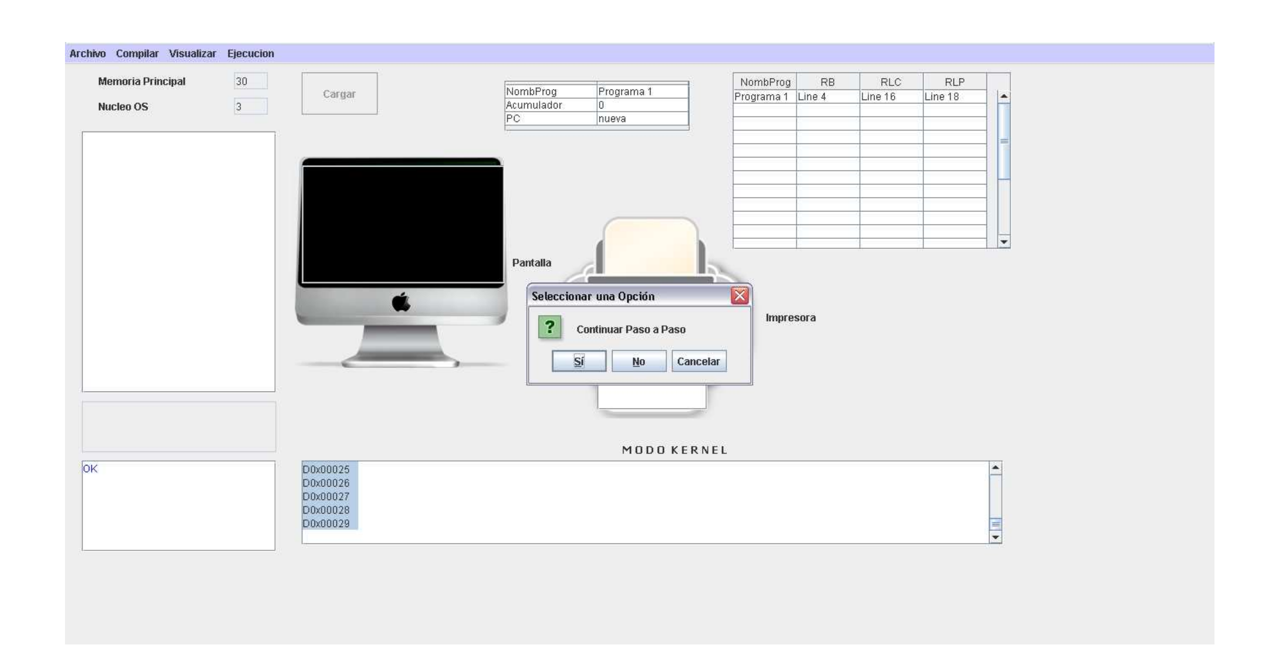
Luego de presentar la dirección, la primera parte de la memoria muestra la posición en donde están las instrucciones, los datos y las variables con los que cuenta el programa. Así:



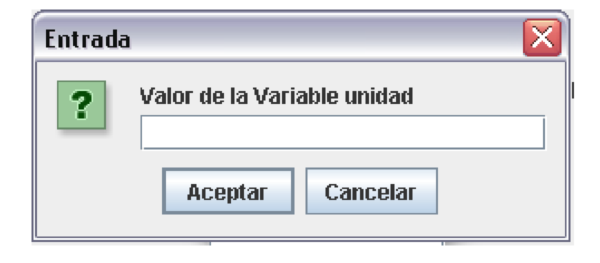
- **EJECUCIÓN DE LOS PROGRAMAS**

Todos los programas se ejecutan en secuencia, sin embargo cuando se encuentran operaciones como **vaya** o **vayasi** se puede ver afectada la secuencia de la ejecución, ya sea de manera condicional o incondicional.

Existen dos formas en las que puede realizarse la ejecución, ya sea de manera continua o paso a paso. En el menú "**Ejecución**" se despliegan las opciones "**Ejecutar**", "**Parar**" y "**Reanudar**" las cuales como indican sus nombres permiten llevar a cabo este proceso permitiendo su ejecución en uno los casos mencionados, la detención y reanudación de la ejecución si fue suspendida en algún momento.



Es posible asignar los valores ingresándolos por teclado (lea) en instrucciones como divida y módulo así:



Cuando se está realizando la ejecución de los programas, es posible ver uno de los espacios el número del programa que está ejecutándose, el valor que va adquiriendo el acumulador y la instrucción que está en ejecución (**PC Program Computer**).

NombProg	Programa 1
Acumulador	9.0
PC	cargue

12. Existe otro espacio que está designado se presentan registros de los programas que se encuentran cargados en la memoria, el número del programa que está ejecutándose, el RB (Registro base), el RLC (Registro límite de código) y el RLP (Registro límite del programa), que incluye además el espacio asignado para las variables.

NombProg	RB	RLC	RLP
Programa 1	Line 4	Line 16	Line 18